

# ΚΕΦΑΛΑΙΟ 2

## *Κύκλος ζωής ανάπτυξης συστήματος*

### *Περιεχόμενα*

- 2.1 Ανάπτυξη Συστήματος
- 2.2 Κύκλος Ζωής Ανάπτυξης Συστήματος
- 2.3 Κλασσικές προβλέψιμες προσεγγίσεις του Κύκλου ζωής Ανάπτυξης Συστήματος
- 2.4 Ευέλικτες – Προσαρμοστικές προσεγγίσεις του Κύκλου ζωής Ανάπτυξης Συστήματος
- 2.5 Στάδια του Κύκλου ζωής Ανάπτυξης Συστήματος
- 2.6 Μεθοδολογίες, μοντέλα, εργαλεία και τεχνικές
- 2.7 Δομημένη Ανάπτυξη Συστήματος (SADT)
- 2.8 Αντικειμενοστραφής Ανάπτυξη Συστήματος
- 2.9 Σύγχρονες Τάσεις στην Ανάπτυξη Συστήματος
  - 2.9.1 Ενοποιημένη διαδικασία (UM)
  - 2.9.2 Ακραίος προγραμματισμός (XM)
  - 2.9.3 Μεθοδολογία Scrum
- Ερωτήσεις – Δραστηριότητες – Θέματα προς Συζήτηση
- Μελέτη περίπτωσης (Σχολική Βιβλιοθήκη)

### *Διδακτικοί στόχοι*

Στόχοι του κεφαλαίου αυτού είναι:

- Να περιγράφουν τα διάφορα στάδια του Κύκλου Ζωής Ανάπτυξης Συστήματος καθώς και να αναγνωρίζουν την συμβολή αυτών στην δημιουργία ενός αποτελεσματικού Πληροφοριακού Συστήματος.
- Να περιγράφουν τις διάφορες προσεγγίσεις του Κύκλου Ζωής Ανάπτυξης Συστήματος.

- Να επιλέγουν και να εφαρμόζουν την κατάλληλη προσέγγιση του Κύκλου Ζωής Ανάπτυξης Συστήματος τόσο σε απλά σενάρια από τον κόσμο των επιχειρήσεων και των οργανισμών όσο και στην καθημερινή τους ζωή.
- Να συγκρίνουν και να αντιπαραβάλλουν διαφορετικές μεθοδολογίες ανάπτυξης συστημάτων και τεχνικών.
- Να εφαρμόζουν αντικειμενοστραφείς και κλασσικές μεθοδολογίες.
- Να περιγράφουν τις σύγχρονες τάσεις στην ανάπτυξη ενός πληροφοριακού συστήματος.

## 2.1 Ανάπτυξη Συστήματος

Την σημερινή εποχή οι υπολογιστές βρίσκονται παντού. Ζούμε σε ένα ψηφιακό κόσμο, όπου η επικοινωνία και η σύνδεση στο διαδίκτυο αποτελούν μέρος της καθημερινότητάς μας. Οι περισσότερες δραστηριότητες μας τόσο σε προσωπικό επίπεδο όσο και σε επαγγελματικό εξαρτώνται από τα microchips, τις γραμμές δικτύου και τις διάφορες εφαρμογές λογισμικού. Μεγαλώνουμε σε ένα κόσμο υψηλής τεχνολογίας. Χρησιμοποιούμε smartphones, laptops, iPads, notepads και εξοπλισμούς ηλεκτρονικών παιχνιδιών. Μέσω κινητών συσκευών λαμβάνουμε μηνύματα, tweets, παρακολουθούμε βίντεο, παίζουμε παιχνίδια και περιηγούμαστε στο διαδίκτυο. Ποιος είναι όμως ο ρόλος των **συστημάτων ανάλυσης και σχεδιασμού** στην ανάπτυξη υψηλής τεχνολογίας λύσεων και εφαρμογών;

Για να απαντήσουμε στην παραπάνω ερώτηση αυτής μελετήσουμε μία παρόμοια κατάσταση, αυτή της κατασκευής κτηρίων. Στο σενάριο αυτό συμμετέχουν ο ιδιοκτήτης/αγοραστής που έχει μία νοερή εικόνα της δημιουργίας του κτηρίου, ο κατασκευαστής που θα κατασκευάσει το κτήριο και ο αρχιτέκτονας που χρησιμεύει ως συνδετικός κρίκος μεταξύ του ιδιοκτήτη και του κατασκευαστή. Ο ρόλος του αρχιτέκτονα είναι να μεταφέρει στον κατασκευαστή εκείνες τις τεχνικές πληροφορίες όπως σχέδια και μακέτες που του είναι απαραίτητες για την κατασκευή του κτηρίου. Οι τεχνικές αυτές πληροφορίες θα προκύψουν μετά από συζήτηση με τον ιδιοκτήτη, κατά την διάρκεια της οποίας ο αρχιτέκτονας θα προσπαθήσει να βοηθήσει τον ιδιοκτήτη να αναπτύξει το όραμα που έχει για το κτήριο. Ο αρχιτέκτονας χρησιμοποιεί διάφορα εργαλεία για να αποτυπώσει τους οραματισμούς του ιδιοκτήτη και στην συνέχεια παραδίδει στον κατασκευαστή σαφείς και ακριβείς οδηγίες για την κατασκευή του κτηρίου σύμφωνα με τις προσδοκίες του ιδιοκτήτη.

Από τα παραπάνω γίνεται σαφές ο σημαντικός ρόλος του αρχιτέκτονα για την επιτυχή κατασκευή του κτιρίου όπως το οραματίστηκε ο ιδιοκτήτης. Ο αρχιτέκτονας έχει να επιτελέσει δύο σημαντικές λειτουργίες. Την ανάλυση και τον σχεδιασμό. Να περιγράψει δηλαδή με λεπτομέρεια το «τι» και το «πώς». «Τι» είναι αυτό το κτήριο που οραματίζεται ο ιδιοκτήτης και «πώς» θα δομηθεί αυτό το σπίτι.

Όπως οι κατασκευαστές όμως δεν αρχίζουν την οικοδόμηση του κτηρίου χωρίς σχέδια, έτσι και οι προγραμματιστές δεν αρχίζουν να προγραμματίζουν, χωρίς να έχουν κάποιον που να έχει αντίστοιχα τον ρόλο του αρχιτέκτονα και ο οποίος θα τους καθοδηγήσει στην δημιουργία του κώδικα μέσω του προσδιορισμού των απαιτήσεων και των περιορισμών του συστήματος. Ο αρχιτέκτονας στο πλαίσιο της ανάπτυξης λογισμικού ονομάζεται **αναλυτής και σχεδιαστής συστημάτων (systems analyst and design)**.

Θα πρέπει να σημειώσουμε ότι η ανάπτυξη ενός πληροφοριακού συστήματος ξεφεύγει από τα στενά όρια του προγραμματισμού και βασίζει την επιτυχία του στην εις βάθος ανάλυση και σχεδιασμό του πληροφοριακού συστήματος. Η δημιουργία ενός συστήματος που βασίζεται σε ελλειπείς ή παρανοημένες απαιτήσεις οδηγεί σε ένα

καθυστερημένο και υπερτιμολογημένο έργο το οποίο δεν επιλύει όλα τα προβλήματα για τα οποία κατασκευάστηκε. Αναπτύσσω ένα πληροφοριακό σύστημα σημαίνει επιλύω προβλήματα με την βοήθεια της τεχνολογίας και όχι μόνο προγραμματίζω. Τι προβλήματα μπορεί να κληθούμε να επιλύσουμε μέσω της ανάπτυξης του κατάλληλου πληροφοριακού συστήματος; Για παράδειγμα, οι πελάτες θέλουν να έχουν την δυνατότητα να παραγγέλνουν προϊόντα όλο το εικοσιτετράωρο. Το πρόβλημα είναι πώς μπορούμε να επεξεργαστούμε αυτές τις παραγγελίες όλο το εικοσιτετράωρο χωρίς να αυξηθεί όμως το κόστος πωλήσεων.

Ο ρόλος λοιπόν του αναλυτή και σχεδιαστή συστημάτων είναι να κατανοήσει τις ανάγκες της επιχείρησης ή του οργανισμού, να συλλάβει το όραμα, να προσδιορίσει την επίλυση του προβλήματος, να επικοινωνήσει το όραμα και την επίλυση, να επιβεβαιώσει ότι η συγκεκριμένη λύση, που υπερκαλύπτει το κόστος ανάπτυξης, ικανοποιεί τις ανάγκες της επιχείρησης ή του οργανισμού και να καθοδηγήσει την αντίστοιχη ομάδα στην υλοποίηση της λύσης. Τα μέλη της ομάδας που συνεργάζονται για την ανάλυση και τον σχεδιασμό του συστήματος πρέπει να κατέχουν τόσο επικοινωνιακές ικανότητες, καθώς θα παίρνουν συνεντεύξεις και θα συνομιλούν με διάφορα μέλη της ομάδας ανάπτυξης όσο και τεχνικές ικανότητες όπως λεπτομερής προσδιορισμός προδιαγραφών και σχεδιασμός λύσεων. Πολλές από τις τεχνικές αυτές ικανότητες σχετίζονται και με την δημιουργία μοντέλων για την αναπαράσταση των προδιαγραφών και της λύσης.

Πιο συγκεκριμένα, η ανάλυση ενός συστήματος περιλαμβάνει εκείνες τις δραστηριότητες που επιτρέπουν στην ομάδα ανάλυσης να κατανοήσει και να προσδιορίσει με ακρίβεια τι ακριβώς πρέπει να πετύχει το νέο σύστημα. Στον παραπάνω ορισμό οι λέξεις κλειδιά είναι «κατανοώ» και «προσδιορίζω». Η ανάλυση ενός συστήματος είναι κάτι πολύ περισσότερο από μια απλή σύντομη δήλωση του προβλήματος. Για παράδειγμα, ένα σύστημα διαχείρισης πελατών θα πρέπει μεταξύ πολλών άλλων περίπλοκων λειτουργιών να παρακολουθεί τους πελάτες, τις εγγραφές των προϊόντων, τις εγγυήσεις και τις διάφορες υπηρεσίες. Η ανάλυση συστήματος περιγράφει λεπτομερώς το "τι" πρέπει να κάνει το προτεινόμενο πληροφοριακό σύστημα για να ικανοποιήσει τις ανάγκες ή να λύσει τα προβλήματα μιας επιχείρησης ή ενός οργανισμού.

Το επόμενο βήμα στην ανάπτυξη ενός συστήματος είναι ο σχεδιασμός. Ο σχεδιασμός ενός συστήματος αποτελείται από εκείνες τις δραστηριότητες που επιτρέπουν στην ομάδα σχεδιασμού να περιγράψει με λεπτομέρεια σε τεχνικό επίπεδο εκείνο το σύστημα που δίνει απάντηση στα υφιστάμενα προβλήματα ή ικανοποιεί τις ανάγκες μιας επιχείρησης ή ενός οργανισμού. Στον παραπάνω ορισμό η λέξη κλειδί είναι «επιλύω». Με άλλα λόγια, ο σχεδιασμός συστημάτων περιγράφει το «πώς» θα λειτουργήσει το σύστημα. Καθορίζει με λεπτομέρεια όλα τα στοιχεία που απαρτίζουν το σύστημα (για παράδειγμα βάσεις δεδομένων, διεπαφές χρηστών, δίκτυα, λογισμικό, λειτουργικές διαδικασίες) και πώς αυτά θα συνεργάζονται μεταξύ τους για την παροχή της επιθυμητής λύσης.

## 2.2. Κύκλος Ζωής Ανάπτυξης Συστήματος

Η ανάπτυξη ενός συστήματος παρουσιάζει ομοιότητες με αυτές ενός project. Δηλαδή οι δραστηριότητες που απαιτούνται για την ανάπτυξη ενός συστήματος πρέπει να προσδιοριστούν, να χρονο-προγραμματιστούν, να οργανωθούν και να είναι υπό συνεχή εποπτεία. Μπορούμε να θεωρήσουμε την ανάπτυξη ενός συστήματος ως ένα προγραμματισμένο εγχείρημα που έχει μια αρχή και ένα τέλος και παράγει κάποιο προκαθορισμένο αποτέλεσμα. Η διαχείριση της ανάπτυξης ενός συστήματος απαιτεί ένα πλαίσιο διαχείρισης έργου για τον καθορισμό και τον συντονισμό όλων των δραστηριοτήτων της ομάδας έργου. Το πλαίσιο αυτό ονομάζεται **Κύκλος Ζωής Ανάπτυξης Συστήματος (ΚΖΑΣ) (Systems Development Life Cycle (SDLC))**.

Ο Κύκλος Ζωής Ανάπτυξης Συστήματος (ΚΖΑΣ) αποτελεί μία από τις βασικές θεμελιώδεις έννοιες στην ανάπτυξη του πληροφοριακών συστημάτων. Τα πληροφοριακά αυτά συστήματα έχουν την δική τους «ζωή» ή τον δικό τους «κύκλο ζωής» και ανάπτυξης. «Ο κύκλος ζωής» ενός πληροφοριακού συστήματος μπορεί να προσεγγίζεται κάθε φορά διαφορετικά σύμφωνα με τις απαιτήσεις του έργου. Οι διαδικασίες όμως που περιλαμβάνονται σε κάθε στάδιο ανάπτυξης ενός πληροφοριακού συστήματος είναι οι ίδιες. Αξίζει να σημειώσουμε το γεγονός ότι κάθε δραστηριότητα που περιλαμβάνεται σε αυτόν τον κύκλο έχει ως σκοπό να παίρνει αυτό που λέει ο πελάτης και να το μετατρέπει σε κάτι που έχει αξία για τον πελάτη. Όλα γίνονται για τον πελάτη. Ας μελετήσουμε τώρα κάθε ένα από τα στάδια αυτά.

Στις περισσότερες περιπτώσεις υπάρχει ένα σύστημα, ακόμα και αν πρόκειται για χαρτί και μολύβι, που το χρησιμοποιεί ο πελάτης για να επιτελέσει την δουλειά του με τον καλύτερο δυνατό τρόπο. Το σύστημα αυτό ο πελάτης θέλει να το βελτιώσει για να αυξήσει την αποδοτικότητα της επιχείρησης ή του οργανισμού. Ξεκινάμε λοιπόν από το **στάδιο μελέτης του έργου** που έχει σκοπό να προσδιορίσει το πεδίο εφαρμογής του νέου συστήματος, να διασφαλίσει την εφικτότητα του σχεδίου, να αναπτύξει το χρονοδιάγραμμα και να υπολογίσει τον προϋπολογισμό του έργου. Στη συνέχεια στο **στάδιο της ανάλυσης**, συζητάμε με τους πελάτες και προσπαθούμε να βρούμε ποιο είναι το πρόβλημα και τι είναι αυτό που ακριβώς χρειάζονται. Ή «τι δουλεύει και τι δεν δουλεύει» στο παλιό σύστημα. Κατά το **στάδιο του σχεδιασμού**, σχεδιάζεται η απάντηση στην ερώτηση, η λύση στο πρόβλημα με βάση τις πληροφορίες που συγκεντρώθηκαν κατά το στάδιο της ανάλυσης. Στη συνέχεια, στο **στάδιο της υλοποίησης** δημιουργείται το πληροφοριακό σύστημα. Στο **στάδιο της υποστήριξης**, βοηθούμε όσους εμπλέκονται με αυτό να το χρησιμοποιήσουν με τον καλύτερο δυνατό τρόπο. Κάθε στάδιο απαιτεί ένα διαφορετικό τρόπο σκέψης, εργασίας και επικοινωνίας με διαφορετικούς ανθρώπους.

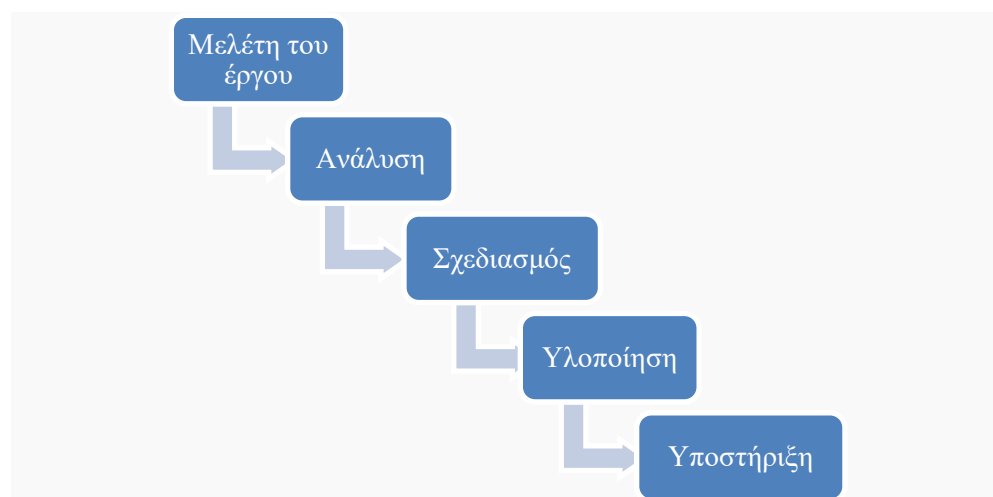
Σε ένα ψηφιακό κόσμο που συνεχώς μεταβάλλεται έχουν αναπτυχθεί και εφαρμοστεί διάφορες προσεγγίσεις ανάπτυξης πληροφοριακών συστημάτων οι οποίες βασίζονται σε διαφορετικούς Κύκλους Ζωής Ανάπτυξης Συστήματος. Οι διαφορετικές αυτές προσεγγίσεις του Κύκλου Ζωής Ανάπτυξης Συστήματος θα μπορούσαν να

ταξινομηθούν με βάση τον βαθμό της προσαρμοστικότητάς τους και της ευελιξίας τους. Στο παρακάτω σχήμα (βλέπε Σχήμα 2.1) στα αριστερά βρίσκονται οι πλήρως προβλέψιμες προσεγγίσεις του Κύκλου Ζωής Ανάπτυξης Συστήματος ενώ στο δεξιό άκρο είναι τοποθετημένες οι πλήρως προσαρμοστικές προσεγγίσεις του Κύκλου Ζωής Ανάπτυξης Συστήματος. Όσο προχωράμε από τα αριστερά προς τα δεξιά τόσο μειώνονται τα στοιχεία της προβλέψιμης συμπεριφοράς και αυξάνονται αυτά της προσαρμοστικής. Στην περίπτωση αυτή αναφερόμαστε σε υβριδικές προσεγγίσεις.



**Σχήμα 2.1:** Η επιλογή του Κύκλου Ζωής Ανάπτυξης Συστήματος εξαρτάται από το έργο

Στην προβλέψιμη προσέγγιση του Κύκλου Ζωής Ανάπτυξης Συστήματος το έργο σχεδιάζεται και οργανώνεται εκ των προτέρων και το νέο πληροφοριακό σύστημα αναπτύσσεται σύμφωνα με το προβλεπόμενο σχέδιο. Για παράδειγμα, μια εταιρεία μπορεί να θέλει να μετατρέψει το παλιό mainframe σύστημα απογραφής της σε ένα νεότερο διαδικτυακό σύστημα τύπου πελάτη/εξυπηρετητή. Σε αυτήν την περίπτωση οι απαιτήσεις είναι πλήρως καθορισμένες και δεν χρειάζεται να προστεθούν νέες διαδικασίες. Χαρακτηριστικό παράδειγμα της πιο προβλέψιμης προσέγγισης αποτελεί το **μοντέλο καταρράκτη**, όπως φαίνεται στο παρακάτω σχήμα (βλέπε Σχήμα 2.2), όπου τα στάδια ανάπτυξης ενός πληροφοριακού συστήματος εξελίσσονται το ένα μετά από το άλλο με την σειρά. Πριν περάσουμε για παράδειγμα στο στάδιο της σχεδίασης πρέπει να έχει ολοκληρωθεί πλήρως το στάδιο της ανάλυσης. Το στάδιο της υλοποίησης δεν μπορεί να ξεκινήσει πριν την αποπεράτωση του σταδίου της σχεδίασης.



**Σχήμα 2.2:** Μοντέλο Καταρράκτη – Κάθε project έχει μία αρχή και ένα τέλος

Στην προσαρμοστική προσέγγιση του Κύκλου Ζωής Ανάπτυξης Συστήματος δεν είναι σαφώς καθορισμένες οι απαιτήσεις ή οι ανάγκες των χρηστών. Αυτό έχει ως αποτέλεσμα να μην μπορεί το έργο να προγραμματιστεί πλήρως εκ των προτέρων. Προσδιορίζεται με σαφήνεια ένα μέρος των απαιτήσεων και στη συνέχεια ακολουθούν κάποιες προκαταρκτικές ενέργειες ανάπτυξης. Η λύση πρέπει να είναι ευέλικτη και να προσαρμόζεται κάθε φορά στην εξέλιξη του έργου. Για παράδειγμα, αρχικά η ομάδα ανάπτυξης εστιάζει σε ένα μικρό αλλά σημαντικό κομμάτι του έργου. Ξεκινάει από την ανάλυση και συνεχίζει με την σχεδίαση και την κατασκευή. Μέσα σε μερικές εβδομάδες παρουσιάζεται στον πελάτη το αποτέλεσμα της συγκεκριμένης εργασίας. Ο πελάτης έχοντας στα χέρια του το πρώτο αλλά σημαντικό δείγμα δουλειάς της ομάδας ανάπτυξης μπορεί να πει «Ναι, αυτό είναι το σύστημα που θέλουμε». Επαναλαμβάνοντας την διαδικασία αυτή αρκετές φορές η ομάδα έργου και προσθέτοντας κάθε φορά περισσότερη λειτουργικότητα στο υπό κατασκευή έργο, μπορεί σε σύντομο χρονικό διάστημα να πλησιάσει στο επιθυμητό αποτέλεσμα σύμφωνα πάντα με τις προδιαγραφές του έργου. Για το λόγο αυτό, η διαδικασία ανάπτυξης ενός σύγχρονου πληροφορικού συστήματος δεν είναι γραμμική αλλά επαναλαμβανόμενα κυκλική όπως φαίνεται στο παρακάτω σχήμα (βλέπε Σχήμα 2.3).



**Σχήμα 2.3:** Κάθε στάδιο επαναλαμβάνεται αρκετές φορές

Πρακτικά, τα περισσότερα συστήματα ανάπτυξης ακολουθούν μία υβριδική προσέγγιση που συνδυάζει στοιχεία και από τις δύο προσεγγίσεις. Είναι και προβλέψιμα και προσαρμοστικά σε διαφορετικό βαθμό για το κάθε ένα. Οι προβλέψιμες προσεγγίσεις είναι οι πιο παραδοσιακές και είναι αυτές που αναπτύχθηκαν πρώτα. Τα μεγάλα συστήματα όμως χρειάζονται ενάμιση με τρία χρόνια ανάπτυξης και στο τέλος το παραγόμενο αποτέλεσμα συνήθως διαφέρει αισθητά από αυτό που είχαν στο νου τους οι πελάτες. Για το λόγο αυτό, η διαδικασία

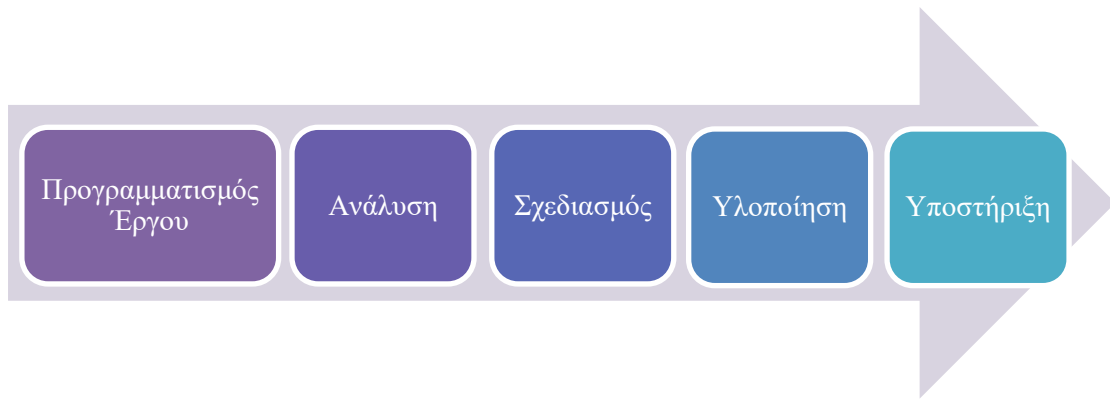
ανάπτυξης ενός σύγχρονου πληροφορικού συστήματος πήρε άλλη μορφή λιγότερο προβλέψιμη και περισσότερο προσαρμοστική και αντικειμενοστραφής με τα επιμέρους στάδια του κύκλου ανάπτυξης να επαναλαμβάνονται κυκλικά είτε αυτά αναφέρονται σε ολόκληρο το έργο είτε σε επιμέρους τμήματά του τα οποία εμπλουτίζονται συνεχώς.

### 2.3. Κλασσικές προβλέψιμες προσεγγίσεις του Κύκλου ζωής Ανάπτυξης Συστήματος

Η ανάπτυξη ενός νέου πληροφοριακού συστήματος συνεπάγεται την εκτέλεση πολλών διαφορετικών δραστηριοτήτων. Οι δραστηριότητες αυτές δεν είναι ανεξάρτητες μεταξύ τους, αλλά συσχετίζονται δηλαδή αποτελούν η μία συνέχεια της άλλης. Στις κλασσικές προβλέψιμες προσεγγίσεις ξεκινάμε αρχικά με τις *δραστηριότητες προγραμματισμού έργου* όπου σχεδιάζουμε, οργανώνουμε και προγραμματίζουμε το έργο. Στην φάση αυτή η επιχείρηση ή ο οργανισμός αναγνωρίζει ότι έχει ένα πρόβλημα να επιλύσει. Οι δραστηριότητες προγραμματισμού έργου χαράζουν τη συνολική δομή του έργου. Στη συνέχεια, μια ομάδα των δραστηριοτήτων που ονομάζονται *δραστηριότητες ανάλυσης* επικεντρώνονται στην κατανόηση του προβλήματος που πρέπει να επιλυθεί καθώς και στον προσδιορισμό των απαιτήσεων. Ο σκοπός είναι να καταλάβουμε ακριβώς τι πρέπει να κάνει το σύστημα για να υποστηρίξει τις λειτουργίες της επιχείρησης ή του οργανισμού. Μια τρίτη ομάδα δραστηριοτήτων εστιάζεται στον σχεδιασμό του νέου συστήματος δηλαδή στον λεπτομερή σχεδιασμό της επίλυσης. Οι δραστηριότητες αυτές, που ονομάζονται *δραστηριότητες σχεδιασμού*, βασίζονται στις απαιτήσεις που έχουν οριστεί νωρίτερα για να σχεδιάσουν τη δομή του προγράμματος και τους αλγόριθμους για το νέο πληροφοριακό σύστημα. Ακολουθούν οι *δραστηριότητες υλοποίησης* που κατασκευάζουν το νέο σύστημα που επιλύει το πρόβλημα. Κύριο μέρος των δραστηριοτήτων υλοποίησης είναι ο προγραμματισμός, ο έλεγχος και η εγκατάσταση του νέου συστήματος.

Αυτές οι τέσσερις παραπάνω ομάδες δραστηριοτήτων του προγραμματισμού, της ανάλυσης, του σχεδιασμού και της υλοποίησης, αναφέρονται συχνά ως **φάσεις (phases)** ή **στάδια** και συνθέτουν το πλαίσιο διαχείρισης του έργου. Μια άλλη φάση, που ονομάζεται *φάση υποστήριξης*, περιλαμβάνει εκείνες τις δραστηριότητες που απαιτούνται για την αναβάθμιση, ενίσχυση και συντήρηση του συστήματος μετά την ολοκλήρωσή του με σκοπό το νέο σύστημα να συνεχίσει να παρέχει τα προβλεπόμενα οφέλη. Η φάση υποστήριξης αποτελεί μέρος του συνολικού Κύκλου Ζωής Ανάπτυξης Συστήματος. Δεν θεωρείται όμως ότι αποτελεί μέρος του αρχικού σχεδίου ανάπτυξης λόγω της φύσης των δραστηριοτήτων που εμπεριέχει. Το παρακάτω σχήμα (βλέπε Σχήμα 2.4) απεικονίζει τα πέντε στάδια του κλασσικού Κύκλου Ζωής Ανάπτυξης Συστήματος.





**Σχήμα 2.4:** Φάσεις Ανάπτυξης Πληροφοριακού Συστήματος

Ένα χαρακτηριστικό παράδειγμα της πιο κλασσικής προβλέψιμης προσέγγισης αποτελεί το μοντέλο καταρράκτη, όπου πριν μεταβούμε από το ένα στάδιο στο επόμενο πρέπει να έχει ολοκληρωθεί πλήρως το προηγούμενο στάδιο. Δεν υπάρχει δυνατότητα επιστροφής στο προηγούμενο στάδιο και εκ νέου αναπροσαρμογή. Το μοντέλο καταρράκτη αποτελεί ένα άκαμπτο σύστημα που προϋποθέτει την έλλειψη ανθρώπινων λαθών. Αν παραλείψουμε να κάνουμε κάτι, τότε δεν μπορούμε να ξαναγυρίσουμε πίσω και να το διορθώσουμε. Από το μοντέλο καταρράκτη οι σύγχρονες προσεγγίσεις ανάπτυξης κρατούν μόνο τις πέντε φάσεις ανάπτυξης αλλά όχι όμως την αυστηρά σειριακή διαδρομή εκτέλεσης αυτών. Προτείνουν ένα τροποποιημένο μοντέλο καταρράκτη με τον κατακερματισμό και την εν μέρει επικάλυψη των πέντε φάσεων του Κύκλου Ζωής Ανάπτυξης Συστήματος, όπως φαίνεται ενδεικτικά στο παρακάτω σχήμα (βλέπε Σχήμα 2.5).



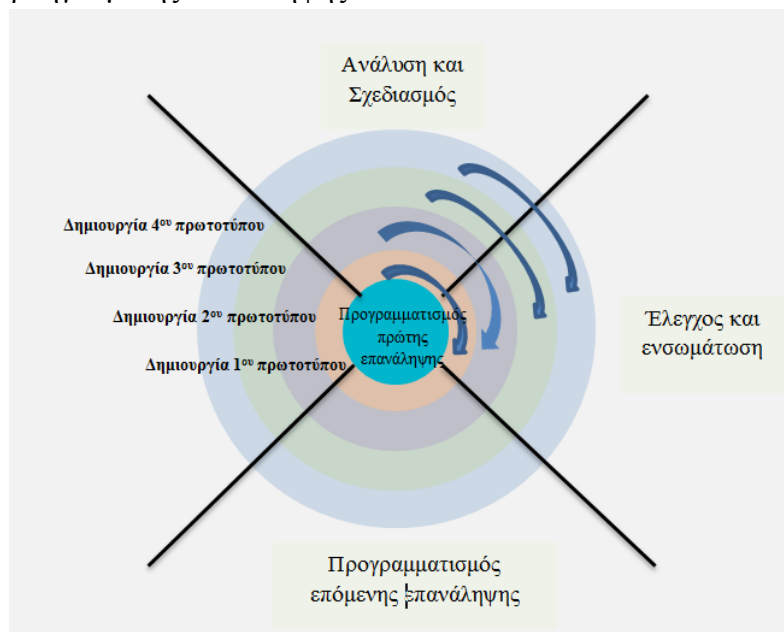
**Σχήμα 2.5:** Επικάλυψη των φάσεων του συστήματος ανάπτυξης

Η επικάλυψη αφορά ένα μέρος μόνο των διαφόρων φάσεων, διότι η μία φάση εξαρτάται από την άλλη. Δεν μπορούμε για παράδειγμα να προχωρήσουμε εις βάθος στον σχεδιασμό χωρίς να έχουμε κατανοήσει πλήρως την φύση του προβλήματος. Άρα, ένα μέρος της ανάλυσης πρέπει να προηγηθεί πρώτα πριν περάσουμε στον σχεδιασμό. Και αυτός ο σχεδιασμός δεν θα είναι πλήρης. Θα αφορά μόνο το αντίστοιχο τμήμα της ανάλυσης που έχει ολοκληρωθεί προηγουμένως.

Από την άλλη πλευρά όμως, λόγω της αλληλεξάρτησης μεταξύ των συνιστωσών του συστήματος, ένα μέρος της ανάλυσης και του σχεδιασμού μπορούν να συμβαδίζουν ταυτόχρονα, όπως συμβαίνει στην περίπτωση της ανάλυσης των αναγκών των χρηστών. Δηλαδή, για να μπορέσουμε να αντιληφθούμε καλύτερα τι θέλουν οι χρήστες από το νέο σύστημα, ο σχεδιασμός ενός μέρους του συστήματος μπορεί να διενεργείται παράλληλα με την ανάλυση.

#### 2.4. Ευέλικτες – Προσαρμοστικές προσεγγίσεις του Κύκλου Ζωής Ανάπτυξης Συστήματος

Σε αντίθεση με τις προβλέψιμες προσεγγίσεις του Κύκλου Ζωής Ανάπτυξης Συστήματος, οι προσαρμοστικές προσεγγίσεις υποθέτουν ότι οι δραστηριότητες που περιλαμβάνονται στην ανάπτυξη του συστήματος πρέπει να αναπροσαρμόζονται κάθε φορά και να ακολουθούν την εξέλιξη του έργου. Αυτό είναι απαραίτητο επειδή οι πτυχές του έργου δεν είναι πλήρως κατανοητές από την αρχή. Μια πρώιμη έκδοση του προσαρμοστικού Κύκλου Ζωής Ανάπτυξης Συστήματος αποτελεί το **σπειροειδές μοντέλο**, όπως φαίνεται στο παρακάτω σχήμα (βλέπε Σχήμα 2.6). Το σπειροειδές μοντέλο είναι μία επαναληπτική διαδικασία που τερματίζει με την ολοκλήρωση του έργου. Σε κάθε επανάληψη κάνουμε τα ίδια πράγματα ξεκινώντας όμως κάθε φορά από διαφορετική βάση και έχοντας ως στόχο να εμπλουτίσουμε το παραγόμενο προϊόν της προηγούμενης επανάληψης.



Σχήμα 2.6: Ο σπειροειδής κύκλος ανάπτυξης

Στο τέλος κάθε κύκλου ή επανάληψης παράγεται ένα πρωτότυπο ως ένα προκαταρκτικό μοντέλο που δείχνει μία πτυχή του συστήματος. Για κάθε πρωτότυπο, η αναπτυξιακή διαδικασία ακολουθεί μια σειριακή διαδρομή μέσα από την ανάλυση, τον σχεδιασμό, την υλοποίηση, τον έλεγχο, την ενσωμάτωση με προηγούμενες συνιστώσες πρωτότυπων και τον προγραμματισμό για το επόμενο πρωτότυπο. Με την ολοκλήρωση του προγραμματισμού για το επόμενο πρωτότυπο ο κύκλος των δραστηριοτήτων αρχίζει ξανά.

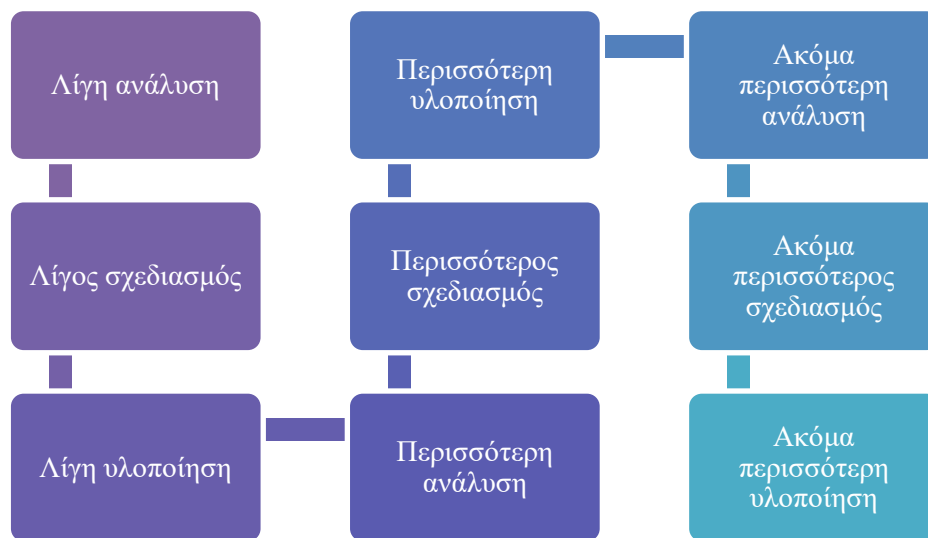
Το παράδειγμα στο Σχήμα 2.6 αποτελεί μία από τις υλοποιήσεις της σπειροειδούς προσέγγισης με την δημιουργία τεσσάρων πρωτοτύπων. Αρχίζει με μια αρχική φάση προγραμματισμού, όπως φαίνεται στο κέντρο της σπείρας. Ο σκοπός της αρχικής αυτής φάσης είναι να συγκεντρωθούν αρκετές πληροφορίες για να μπορέσει να ξεκινήσει η ανάπτυξη του πρώτου πρωτότυπου. Η αρχική φάση προγραμματισμού των δραστηριοτήτων περιλαμβάνει την μελέτη σκοπιμότητας, την έρευνα σχετικά με τις απαιτήσεις των χρηστών, την παραγωγή εναλλακτικών προτάσεων εφαρμογής καθώς και την επιλογή του συνολικού σχεδιασμού και της στρατηγικής υλοποίησης. Μετά τον αρχικό προγραμματισμό η ομάδα ανάπτυξης αναλαμβάνει την δημιουργία του πρώτου πρωτότυπου και ακολουθούν στη συνέχεια και τα υπόλοιπα πρωτότυπα μέχρι την ολοκλήρωση του έργου.

Κατά την εξέλιξη του έργου εμφανίζονται μία σειρά από κίνδυνοι οι οποίοι αν δεν αντιμετωπιστούν έγκαιρα είναι δυνατό να τερματίσουν την επιτυχή έκβαση του έργου. Σε κάθε κύκλο ανάπτυξης ή επανάληψης καταγράφονται οι κίνδυνοι που εμπεριέχουν οι διαδικασίες και οι εναλλακτικές λύσεις όπου υπάρχουν. Κατά την πρώτη επανάληψη εστιάζουμε σε εκείνον τον τομέα ο οποίος φαίνεται να παρουσιάζει τους μεγαλύτερους κινδύνους, όπως για παράδειγμα η εφικτότητα της εφαρμογής της νέας τεχνολογίας. Στην περίπτωση αυτή η πρώτη επανάληψη θα μπορούσε να επικεντρωθεί σε ένα πρωτότυπο που αποδεικνύει ότι η συγκεκριμένη τεχνολογία μπορεί να λειτουργήσει έτσι όπως έχει σχεδιαστεί. Στη συνέχεια η δεύτερη επανάληψη θα μπορούσε να εστιαστεί στην δημιουργία ενός πρωτότυπου που θα διευθετεί τους κινδύνους που σχετίζονται με τις απαιτήσεις του συστήματος. Για μία άλλη εφαρμογή ο μεγαλύτερος κίνδυνος θα μπορούσε να ήταν η αποδοχή της αλλαγής εκ μέρους των χρηστών. Έτσι, η πρώτη επανάληψη θα μπορούσε να επικεντρωθεί στην δημιουργία ενός πρωτότυπου που θα αποδεικνύει την βελτίωση του ψηφιακού εργασιακού περιβάλλοντος με την εφαρμογή του νέου συστήματος.

Στο σπειροειδές μοντέλο χρησιμοποιείται ο όρος επανάληψη. Στην επίλυση προβλημάτων, οι επαναλήψεις χρησιμοποιούνται για να διαιρέσουμε ένα πολύ μεγάλο και σύνθετο πρόβλημα σε απλούστερα υποπροβλήματα τα οποία μπορούμε να διαχειριστούμε πιο εύκολα. Η επίλυση κάθε υποπροβλήματος οδηγεί στην επίλυση του αρχικού προβλήματος.

Η ανάπτυξη πληροφοριακών συστημάτων χρησιμοποιεί την επανάληψη για τον ίδιο σκοπό. Παίρνουμε ένα μεγάλο σύστημα και το κατακερματίζουμε σε μικρότερες συνιστώσες. Στη συνέχεια προγραμματίζουμε, αναλύουμε, σχεδιάζουμε και

υλοποιούμε κάθε μικρότερη συνιστώσα. Το τελευταίο βήμα είναι αυτό της ολοκλήρωσης όπου συνδυάζουμε τις μικρότερες συνιστώσες σε μια ολοκληρωμένη λύση. Η προσέγγιση αυτή συχνά ονομάζεται **επαναληπτική προσέγγιση (iterative approach)** του Κύκλου Ζωής Ανάπτυξης Συστήματος. Πολλές από τις πιο δημοφιλείς προσαρμοστικές προσεγγίσεις χρησιμοποιούν σήμερα την επανάληψη ως το θεμελιώδες στοιχείο της προσέγγισης. Το σχήμα 2.7 απεικονίζει πώς λειτουργεί μια επαναληπτική προσέγγιση.



**Σχήμα 2.7:** Οι δραστηριότητες ανάπτυξης συστήματος επαναλαμβάνονται

Επανάληψη σημαίνει ότι οι δραστηριότητες του έργου ανάπτυξης, όπως η ανάλυση, ο σχεδιασμός και η υλοποίηση επαναλαμβάνονται συνεχώς. Ο αριθμός των επαναλήψεων εξαρτάται από την πολυπλοκότητα του έργου. Με κάθε επανάληψη, τα μέλη της ομάδας ανάπτυξης βελτιώνουν το αποτέλεσμα, έτσι ώστε να είναι πιο κοντά σε αυτό που τελικά χρειαζόμαστε. Η επανάληψη βασίζεται στην παραδοχή ότι δεν είναι δυνατό να πετύχουμε το σωστό αποτέλεσμα με την πρώτη φορά. Αρχικά η ανάλυση, ο σχεδιασμός και η υλοποίηση είναι μικρής έκτασης. Τα πρώτα αποτελέσματα όμως που παίρνουμε μας δίνουν την δυνατότητα να γνωρίζουμε αν πραγματικά το σύστημα θα λειτουργήσει και θα πετύχει τους στόχους του. Στη συνέχεια αναλύοντας, σχεδιάζοντας και υλοποιώντας κάθε φορά όλο και μεγαλύτερο τμήμα μπορούμε να κάνουμε βελτιώσεις. Όσο προχωράει η επαναληπτική αυτή διαδικασία τόσο πλησιάζουμε όλο και πιο κοντά στους στόχους που έχουμε θέσει και ταυτόχρονα μειώνουμε τους κινδύνους που μπορούν να εμφανιστούν κατά την ανάπτυξη όπως είναι η αλλαγή της τεχνολογίας.

Μπορούμε να οργανώσουμε τις επαναλήψεις με διάφορους τρόπους. Μία προσέγγιση είναι να ορίσουμε τις πιο βασικές λειτουργίες που πρέπει το σύστημα να περιλαμβάνει και στη συνέχεια να τις υλοποιήσουμε στην πρώτη επανάληψη. Με την ολοκλήρωση των βασικών αυτών λειτουργιών του συστήματος υλοποιούμε στη συνέχεια τις επόμενες και λιγότερο κρίσιμες λειτουργίες του συστήματος. Τέλος, οι

προαιρετικές λειτουργίες του συστήματος, για τις οποίες λέμε «καλό είναι να υπάρχουν» υλοποιούνται στην τελευταία επανάληψη. Μια άλλη προσέγγιση είναι να επικεντρωνόμαστε κάθε φορά σε ένα υποσύστημα. Η υλοποίηση του πρώτου υποσυστήματος περιέχει βασικές λειτουργίες και τα δεδομένα από τα οποία εξαρτώνται τα άλλα υποσυστήματα. Στη συνέχεια, η επόμενη επανάληψη περιλαμβάνει ένα πρόσθετο υποσύστημα και ούτω καθεξής.

Μερικές φορές οι επαναλήψεις ορίζονται σύμφωνα με την πολυπλοκότητα ή τον κίνδυνο ορισμένων συνιστωσών. Συχνά, οι πιο σύνθετες ή υψηλού κινδύνου συνιστώσες του συστήματος διευθετούνται πρώτα για να μπορέσουν οι στόχοι και οι αρχικές απαιτήσεις να αλλάξουν έγκαιρα χωρίς τεράστιες συνέπειες για την ανάπτυξη του έργου. Άλλες φορές, μερικά από τα απλούστερα συστατικά του συστήματος αντιμετωπίζονται πρώτα για να έχουμε όσο είναι δυνατό γρηγορότερα μεγαλύτερο μέρος του συστήματος ολοκληρωμένο. Το πώς οι επαναλήψεις ορίζονται, εξαρτάται από πολλούς παράγοντες και μπορεί να είναι διαφορετικές κάθε φορά σύμφωνα με το υπό ανάπτυξη έργο. Οι περισσότερες προσαρμοστικές προσεγγίσεις δίνουν προτεραιότητα στην αντιμετώπιση των πιο δύσκολων προβλημάτων με τον υψηλότερο κίνδυνο.

Μια σχετική προσέγγιση, η οποία είναι ένας τύπος επαναληπτικής προσέγγισης είναι αυτή της **αυξητικής ανάπτυξης (incremental development)**. Με την προσέγγιση αυτή, ολοκληρώνονται μέρη του συστήματος σε λίγες επαναλήψεις και τίθεται σε λειτουργία το σύστημα για τους χρήστες έτσι ώστε να μπορούν να επωφεληθούν από αυτό άμεσα. Στη συνέχεια με την μέθοδο των επαναλήψεων αναπτύσσεται ένα άλλο μέρος του συστήματος, το οποίο διασυνδέεται με το πρώτο μέρος και τίθεται ο τελικός συνδυασμός πάλι σε λειτουργία. Τέλος, ολοκληρώνεται το τελευταίο μέρος του συστήματος και ενσωματώνεται με τα υπόλοιπα. Σήμερα, οι περισσότερες διαδικασίες ανάπτυξης συστήματος χρησιμοποιούν ποικίλους βαθμούς επανάληψης. Μία από αυτές που περιγράφεται στην συνέχεια είναι η αντικειμενοστραφής προσέγγιση που χαρακτηρίζεται ως άκρως επαναληπτική.

## 2.5. Στάδια του Κύκλου ζωής Ανάπτυξης Συστήματος

Στις προηγούμενες ενότητες περιγράψαμε γενικά τις διάφορες φάσεις του Κύκλου Ζωής Ανάπτυξης Συστήματος που είναι ο προγραμματισμός, η ανάλυση, ο σχεδιασμός, η υλοποίηση και η υποστήριξη και εξηγήσαμε πώς οι δραστηριότητες της κάθε φάσης διενεργούνται επαναληπτικά. Στη συνέχεια, θα παρουσιάσουμε αναλυτικά τις δραστηριότητες της κάθε φάσης του Κύκλου Ζωής Ανάπτυξης Συστήματος.

### Προγραμματισμός έργου

Οι πρωταρχικοί στόχοι του προγραμματισμού έργου είναι να προσδιορίσουμε το πεδίο εφαρμογής του νέου συστήματος, να διασφαλίσουμε ότι το έργο αυτό είναι εφικτό και υλοποιήσιμο, να σχεδιάσουμε τον χρονοπρογραμματισμό και τους πόρους

που θα χρειαστούμε και τέλος να κοστολογήσουμε το έργο. Οι δραστηριότητες που περιλαμβάνονται στον προγραμματισμό του έργου είναι οι παρακάτω:

- Καθορισμός του προβλήματος
- Χρονοπρογραμματισμός του έργου
- Επιβεβαίωση δυνατότητας υλοποίησης του έργου
- Στελέχωση του έργου
- Αρχικοποίηση του έργου

Η πιο σημαντική δραστηριότητα του προγραμματισμού του έργου είναι ο ακριβής προσδιορισμός του προβλήματος και το πεδίο εφαρμογής της λύσης του προβλήματος. Σε αυτό το στάδιο του έργου, δεν γνωρίζουμε όλες τις λειτουργίες ή τις διαδικασίες που θα πρέπει να συμπεριληφθούν στο σύστημα. Ωστόσο, είναι σημαντικό να εντοπίσουμε τις κύριες χρήσεις του νέου συστήματος και τα προβλήματα της επιχείρησης ή του οργανισμού που θα κληθεί το νέο σύστημα να αντιμετωπίσει.

### **Δραστηριότητες Ανάλυσης**

Ο πρωταρχικός στόχος των δραστηριοτήτων ανάλυσης είναι να κατανοήσουμε και να τεκμηριώσουμε τις ανάγκες και τις απαιτήσεις του νέου συστήματος. Η ανάλυση είναι ουσιαστικά μια ανακαλυπτική διαδικασία. Οι λέξεις κλειδιά που χαρακτηρίζουν τις δραστηριότητες της ανάλυσης είναι η *ανακάλυψη* και η *κατανόηση*. Οι δραστηριότητες που περιλαμβάνονται σε αυτό το στάδιο είναι οι παρακάτω:

- Συλλογή πληροφοριών
- Προσδιορισμός απαιτήσεων συστήματος
- Κατασκευή πρωτοτύπων για την διερεύνηση των απαιτήσεων
- Ιεράρχηση απαιτήσεων
- Δημιουργία και αξιολόγηση εναλλακτικών λύσεων
- Επιλογή εναλλακτικής λύσης

### **Δραστηριότητες σχεδιασμού**

Ο στόχος των δραστηριοτήτων σχεδιασμού είναι να σχεδιάσουμε το σύστημα με βάση τις καθορισμένες απαιτήσεις και τις αποφάσεις που έχουν ληφθεί κατά τη διάρκεια της ανάλυσης. Ο υψηλού επιπέδου σχεδιασμός συνίσταται στην ανάπτυξη μια αρχιτεκτονικής δομής για τα στοιχεία του λογισμικού, των βάσεων δεδομένων, των διεπαφών χρήστη και του λειτουργικού περιβάλλοντος. Ο σχεδιασμός χαμηλού επιπέδου συνεπάγεται την ανάπτυξη των λεπτομερών αλγορίθμων και δομών δεδομένων που είναι απαραίτητα για την ανάπτυξη λογισμικού. Επτά κύριες δραστηριότητες πρέπει να ολοκληρωθούν κατά τη διάρκεια της φάσης σχεδιασμού:

- Σχεδιασμός και ολοκλήρωση του δικτύου
- Σχεδιασμός της αρχιτεκτονικής της εφαρμογής

- Σχεδιασμός των διεπαφών χρήστη
- Σχεδιασμός των διασυνδέσεων συστήματος
- Σχεδιασμός και ενσωμάτωση της βάσης δεδομένων
- Δημιουργία πρωτοτύπου
- Σχεδιασμός και ενσωμάτωση των ελέγχου συστήματος

### **Δραστηριότητες Υλοποίησης**

Με τις δραστηριότητες υλοποίησης παραδίδεται το τελικό σύστημα και είναι έτοιμο να χρησιμοποιηθεί. Στόχος δεν είναι μόνο να παραχθεί ένα αξιόπιστο, πλήρως λειτουργικό πληροφοριακό σύστημα, αλλά και να διασφαλίσει ότι οι όλοι χρήστες του συστήματος έχουν εκπαιδευτεί και η επιχείρηση ή ο οργανισμός είναι έτοιμοι να επωφεληθούν, όπως αναμενόταν από τη χρήση του συστήματος. Όλες οι προηγούμενες δραστηριότητες συγκλίνουν και οδηγούν στην δημιουργία ενός λειτουργικού πληροφοριακού συστήματος. Πέντε είναι οι κύριες δραστηριότητες που συνθέτουν το στάδιο υλοποίησης:

- Κατασκευή συνιστωσών λογισμικού
- Έλεγχος και δοκιμή
- Μετατροπή δεδομένων
- Εκπαίδευση χρηστών και τεκμηρίωση συστήματος
- Εγκατάσταση συστήματος

### **Δραστηριότητες Υποστήριξης**

Ο στόχος των δραστηριοτήτων υποστήριξης είναι να διατηρήσουν το σύστημα σε πλήρη λειτουργία στο πέρασμα του χρόνου μετά την αρχική εγκατάσταση του. Το νέο σύστημα εγκαθίσταται και τίθεται σε λειτουργία. Από το σημείο εκείνο ξεκινούν οι δραστηριότητες υποστήριξης οι οποίες διαρκούν όσο και η «παραγωγική ζωή» του συστήματος. Οι περισσότερες επιχειρήσεις ή οργανισμοί προσδοκούν η «παραγωγική ζωή» του νέου συστήματος να έχει μεγάλη διάρκεια. Κατά την διάρκεια της υποστήριξης, αναβαθμίσεις ή βελτιώσεις είναι δυνατόν να διενεργηθούν με σκοπό την επέκταση των δυνατοτήτων του συστήματος. Αυτές οι αναβαθμίσεις ή βελτιώσεις αναπτύσσονται σύμφωνα με τον Κύκλο Ζωής Ανάπτυξης Συστήματος. Οι τρεις κύριες δραστηριότητες που λαμβάνουν χώρα κατά τη διάρκεια της υποστήριξης είναι οι παρακάτω:

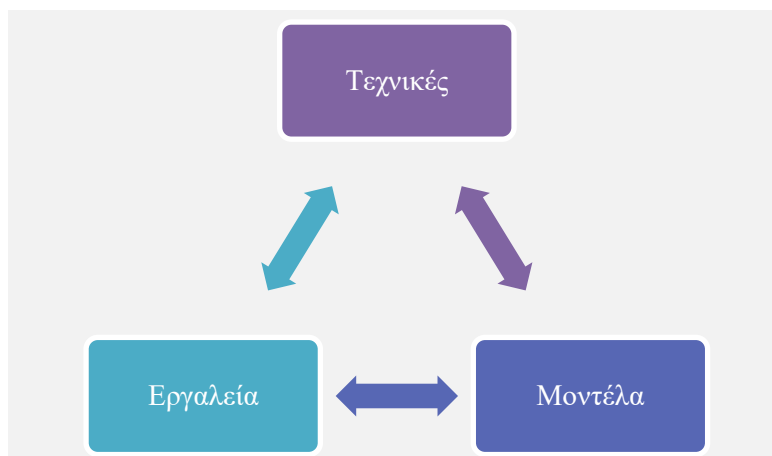
- Διατήρηση του συστήματος
- Βελτίωση του συστήματος
- Υποστήριξη χρηστών

## **2.6. Μεθοδολογίες, μοντέλα, εργαλεία και τεχνικές**

Οι αναλυτές συστημάτων έχουν μια ποικιλία από βοηθήματα στην διάθεσή τους που τους βοηθούν να ολοκληρώσουν τις δραστηριότητές και τα καθήκοντα τους. Μεταξύ αυτών είναι οι μεθοδολογίες, τα μοντέλα, τα εργαλεία και οι τεχνικές.

### Μεθοδολογίες

Μια μεθοδολογία ανάπτυξης συστήματος παρέχει οδηγίες για την ολοκλήρωση της κάθε δραστηριότητας στον Κύκλο Ζωής Ανάπτυξης Συστήματος και περιλαμβάνει συγκεκριμένες τεχνικές, εργαλεία και μοντέλα τα οποία συνδέονται μεταξύ τους όπως φαίνεται στο παρακάτω σχήμα (βλέπε Σχήμα 2.8).



*Σχήμα 2.8: Σχέσεις μεταξύ συστατικών μιας μεθοδολογίας*

Επειδή μια μεθοδολογία περιέχει οδηγίες για το πώς θα χρησιμοποιήσουμε τα μοντέλα, τα εργαλεία και τις τεχνικές, ας προσπαθήσουμε να καταλάβουμε πρώτα τι είναι τα μοντέλα, τα εργαλεία και οι τεχνικές. Τα μοντέλα, τα εργαλεία και οι τεχνικές είναι αυτά που θα σας βοηθήσουν να διαχειριστείτε αποτελεσματικά το έργο που έχετε αναλάβει.

### Μοντέλα (Models)

Κάθε φορά που θέλουμε να καταγράψουμε ή να επικοινωνήσουμε πληροφορίες σε οποιοδήποτε πλαίσιο, είναι πολύ χρήσιμο να δημιουργήσουμε ένα μοντέλο. Ένα μοντέλο στην ανάπτυξη πληροφοριακών συστημάτων έχει τον ίδιο σκοπό με οποιοδήποτε άλλο μοντέλο. Ένα μοντέλο είναι μια αναπαράσταση μιας σημαντικής πτυχής του πραγματικού κόσμου. Μερικές φορές ο όρος **αφαίρεση - γενίκευση (abstraction)** χρησιμοποιείται διότι διαχωρίζουμε από το υπόλοιπο σύνολο αυτό που έχει την μεγαλύτερη σημασία για εμάς και αυτό είναι που κρατάμε. Ας θεωρήσουμε το μοντέλο ενός αεροπλάνου. Αν θέλουμε να μιλήσουμε για την αεροδυναμική του αεροπλάνου, είναι χρήσιμο να έχουμε ένα μικρό μοντέλο που δείχνει το συνολικό σχήμα του αεροπλάνου στις τρεις διαστάσεις. Μερικές φορές ένα γράφημα που δείχνει λεπτομέρειες από την διατομή της πτέρυγας του αεροπλάνου είναι αυτό που χρειαζόμαστε. Σε άλλες περιπτώσεις, μια λίστα των μαθηματικών χαρακτηριστικών



του αεροπλάνου θα μπορούσε να χρησιμοποιηθεί για να κατανοήσουμε την συμπεριφορά του αεροπλάνου. Όλα αυτά είναι μοντέλα του ίδιου αεροπλάνου.

Ορισμένα μοντέλα είναι παρόμοια με το πραγματικό προϊόν. Άλλα μοντέλα αποτελούν γραφικές αναπαραστάσεις σημαντικών λεπτομερειών ή είναι αφηρημένοι μαθηματικοί συμβολισμοί. Κάθε μοντέλο δίνει έμφαση σε ένα διαφορετικό είδος των πληροφοριών. Στο σχεδιασμό του αεροπλάνου, οι μηχανικοί αεροσκαφών χρησιμοποιούν πολλά διαφορετικά μοντέλα. Το να είναι κάποιος μηχανικός αεροσκαφών σημαίνει πως γνωρίζει πώς να δημιουργεί και να χρησιμοποιεί όλα αυτά τα μοντέλα. Το ίδιο ακριβώς ισχύει και με τα μέλη της ομάδας ανάπτυξης. Τα μοντέλα για τα πληροφοριακά συστήματα όμως δεν είναι τόσο τυποποιημένα ή ακριβή. Αυτό οφείλεται στο γεγονός ότι ο τομέας που μελετάει τα μοντέλα των πληροφοριακών συστημάτων γνωρίζει ανάπτυξη τα τελευταία χρόνια καθώς και στο ότι τα πληροφοριακά συστήματα έχουν να κάνουν με καταστάσεις και έννοιες που δεν έχουν την φυσική υπόσταση των αεροπλάνων.

Τι είδους μοντέλα όμως χρησιμοποιούνται στα πληροφοριακά συστήματα; Τα μοντέλα που χρησιμοποιούνται στην ανάπτυξη συστημάτων περιλαμβάνουν αναπαραστάσεις εισόδων εξόδων, διαδικασιών, δεδομένων, αντικείμενων, αλληλεπιδράσεων μεταξύ αντικείμενων, τοποθεσιών, δικτύων και συσκευών μεταξύ άλλων. Τα περισσότερα από τα αυτά είναι γραφικά μοντέλα δηλαδή *διαγράμματα* και *γραφήματα*. Ένα άλλο είδος σημαντικού μοντέλου που αναπτύσσουμε και χρησιμοποιούμε είναι αυτό του σχεδιασμού ή προγραμματισμού του έργου. Στο μοντέλο αυτό παρουσιάζονται οι εργασίες που πρέπει να διεκπεραιωθούν καθώς και οι αντίστοιχες ημερομηνίες ολοκλήρωσης. Στα παραπάνω μοντέλα προστίθεται και το διάγραμμα που δείχνει το σύνολο των ατόμων που ασχολούνται με το έργο.

### **Εργαλεία (Tools)**

Ένα εργαλείο στο πλαίσιο της ανάπτυξης του πληροφοριακού συστήματος είναι ένα υποστηρικτικό λογισμικό που βοηθά στη δημιουργία μοντέλων ή άλλα συνιστωσών που είναι απαραίτητες για την ανάπτυξη του έργου. Τα εργαλεία μπορεί να είναι απλά σχεδιαστικά προγράμματα για τη δημιουργία διαγραμμάτων. Θα μπορούσαν να περιλαμβάνουν μια εφαρμογή βάσης δεδομένων που να αποθηκεύει πληροφορίες αναφορικά με το έργο, όπως είναι οι ορισμοί ροών δεδομένων ή οι γραπτές περιγραφές των διαδικασιών. Ένα λογισμικό διαχείρισης έργου είναι ένα άλλο παράδειγμα εργαλείου που χρησιμοποιείται για να δημιουργήσει μοντέλα για τα καθήκοντα και τις εξαρτήσεις μεταξύ των καθηκόντων αυτών.

Τα εργαλεία έχουν σχεδιαστεί ειδικά για να βοηθήσουν τα μέλη της ομάδας ανάπτυξης του συστήματος. Οι προγραμματιστές θα πρέπει να είναι εξοικειωμένοι με τα ολοκληρωμένα περιβάλλοντα ανάπτυξης (integrated development environments - IDEs) που περιλαμβάνουν πολλά εργαλεία για την υποστήριξη του προγραμματισμού, όπως είναι οι έξυπνοι συντάκτες, η θεματική βοήθεια και τα εργαλεία εκσφαλμάτωσης. Μερικά εργαλεία μπορούν να δημιουργήσουν κώδικα

προγράμματος για τους προγραμματιστές ή να δημιουργήσουν μοντέλα από κώδικα προγράμματος έτσι ώστε ο προγραμματιστής να μπορεί να προσδιορίσει τι κάνει το πρόγραμμα σε περίπτωση που ο φάκελος τεκμηρίωσης απουσιάζει. Εργαλεία οπτικής μοντελοποίησης είναι διαθέσιμα στους αναλυτές συστημάτων για να τους βοηθήσουν να δημιουργήσουν και να επαληθεύσουν μοντέλα συστημάτων καθώς και να δημιουργήσουν κώδικα προγράμματος απευθείας από μοντέλα.

### Τεχνικές (Techniques)

Μια τεχνική στην ανάπτυξη συστήματος είναι μια συλλογή από κατευθυντήριες γραμμές που υποστηρίζουν τον αναλυτή στην προσπάθειά του να ολοκληρώσει μια δραστηριότητα. Μια τεχνική μπορεί να περιλαμβάνει οδηγίες βήμα-προς-βήμα για τη δημιουργία ενός μοντέλου ή γενικές συμβουλές για τη συλλογή πληροφοριών από τους χρήστες του συστήματος. Συναντάμε τεχνικές μοντελοποίησης δεδομένων, ελέγχου λογισμικού, συνέντευξης καθώς και σχεδιασμού σχεσιακών βάσεων δεδομένων.

### 2.7. Δομημένη Ανάπτυξη Συστήματος (Structured Analysis and Design Technique - SADT)

Η δομημένη ανάλυση, η δομημένη σχεδίαση και ο δομημένος προγραμματισμός είναι τρεις τεχνικές που αποτελούν την δομημένη ανάπτυξη. Πολλές φορές αυτές οι τεχνικές αναφέρονται με τα αρχικά των λέξεων **Structured Analysis and Design Technique (SADT)**.

Οι τεχνικές του **δομημένου προγραμματισμού** αναπτύχθηκαν στην δεκαετία του 60 και αποτέλεσαν μια προσπάθεια για την παροχή οδηγιών που θα βελτιώναν την ποιότητα των προγραμμάτων. Οι τεχνικές της **δομημένης σχεδίασης** αναπτύχθηκαν την δεκαετία του 70, για να συνδεθούν διαφορετικά προγράμματα σε ένα πληροφοριακό σύστημα, καθώς παρείχαν οδηγίες για το είδος των προγραμμάτων και την ιεράρχησή τους. Για το σκοπό αυτό αναπτύχθηκαν τα **Διαγράμματα Δομής (ΔΔ)**. Οι τεχνικές της δομημένης ανάλυσης εξελίχθηκαν στις αρχές της δεκαετίας του 1980 για να διευκρινιστούν οι απαιτήσεις ενός πληροφοριακού συστήματος, πριν οι αναλυτές σχεδιάσουν τα προγράμματα. Τα **Διαγράμματα Ροής Δεδομένων (ΔΡΔ)** αποτελούν τη διαγραμματική αποτύπωση των απαιτήσεων του συστήματος. Από τη φάση της ανάλυσης τα ΔΡΔ μετατρέπονται σε ΔΔ για να χρησιμοποιηθούν στη Σχεδίαση.

Στη δεκαετία του 80, αναπτύχθηκαν τεχνικές σχεδίασης αρχείων και βάσεων δεδομένων. Τα Συστήματα Διαχείρισης βάσεων Δεδομένων χρησιμοποιούνται στην δομημένη σχεδίαση με σκοπό την αλληλεπίδραση των προγραμμάτων με τη βάση δεδομένων και για αυτό το λόγο αναπτύχθηκαν τα **Διαγράμματα Οντοτήτων Συσχετίσεων (ΔΟΣ)**. Επιπρόσθετα, κατά την δεκαετία του 80, αναπτύχθηκαν τεχνικές σχεδίασης διεπαφής χρήστη (για παράδειγμα σχεδιασμός menus).

Σύμφωνα με τη δομημένη προσέγγιση το Πληροφοριακό Σύστημα θεωρείται ότι αποτελεί μια ενιαία και σχετικά αυτοτελή, μεγάλη και σύνθετη λειτουργική μονάδα. Πλεονεκτήματα της δομημένης ανάπτυξης αποτελεί η ανεξαρτησία των δεδομένων από το λογισμικό, η ευκολία στην κατανόηση του κώδικα και η συμβατότητα με τον δομημένο προγραμματισμό. Χαρακτηριστικό της δομημένης ανάπτυξης συστήματος είναι η αυστηρή πειθαρχία στην οργάνωση του κώδικα του προγράμματος και η σειριακή εκτέλεση των φάσεων ανάπτυξης που όμως πολλές φορές οδηγεί σε καθυστερήσεις στην υλοποίηση του έργου.

### **Μειονεκτήματα της δομημένης προσέγγισης**

Η δομημένη ανάπτυξη ενός συστήματος με την πάροδο του χρόνου εξελίχθηκε σε αρκετές παραλλαγές. Οι αναλυτές που έμειναν στην αρχικές εκδόσεις δεν χρησιμοποίησαν πολλές βελτιώσεις της, ενώ κάποιοι χρησιμοποίησαν τμήματα των νέων τεχνικών χωρίς να τις κατανοήσουν σε όλη τους την έκταση και να τις μάθουν με όλες τις λεπτομέρειες. Πολλοί πιστεύουν ότι η δομημένη προσέγγιση έχει πολλές αδυναμίες καθώς οι τεχνικές που χρησιμοποιούνται δεν καλύπτουν όλες τις ενέργειες της ανάλυσης και σχεδίασης. Επίσης, πολλοί αναλυτές θεωρούν ότι η μετάβαση από τα Διαγράμματα Ροής Δεδομένων (ΔΡΔ) (δομημένη ανάλυση) στα Διαγράμματα Δομής (δομημένη σχεδίαση) στην πράξη δεν λειτουργεί. Μεγάλο μειονέκτημα, αποτελούν η πολύ αργή ανάπτυξη (κρατά μερικά χρόνια για μεγάλα Π.Σ.) γιατί έχει σειριακό χαρακτήρα (πρέπει πρώτα να τελειώσει η ανάλυση για να αρχίσει η σχεδίαση, κλπ) και οι αλλαγές στις επιχειρηματικές απαιτήσεις που αποτυπώνονται με μεγάλη δυσκολία και δεν επιτρέπουν την επαναχρησιμοποίηση τμημάτων λογισμικού (reuse).

### **2.8. Αντικειμενοστραφής ανάπτυξη συστήματος**

Μια εντελώς διαφορετική προσέγγιση στην ανάπτυξη πληροφοριακών συστημάτων αποτελεί η αντικειμενοστραφής προσέγγιση. Το σύστημα εξετάζεται ως συλλογή από αλληλεπιδρώμενα αντικείμενα τα οποία συνεργάζονται για την πραγματοποίηση ενός έργου. Το σύστημα αποτελείται από αντικείμενα που μπορούν να απαντούν σε μηνύματα. Αυτό οδήγησε στην ανάπτυξη διαφορετικών προσεγγίσεων στην ανάλυση, στον σχεδιασμό και στον προγραμματισμό των συστημάτων. Η προσέγγιση αυτή αναπτύχθηκε όταν στη δεκαετία του 70 άρχισαν να αναπτύσσονται οι αντικειμενοστραφείς γλώσσες προγραμματισμού.

Επειδή η αντικειμενοστραφής προσέγγιση μελετά τα πληροφοριακά συστήματα ως συλλογές από αντικείμενα που αλληλεπιδρούν, η αντικειμενοστραφής ανάλυση (**Object-Oriented Analysis (OOA)**) καθορίζει όλα τα είδη των αντικειμένων που αλληλεπιδρούν με το σύστημα και μελετά τις αλληλεπιδράσεις χρήστη που απαιτούνται για την ολοκλήρωση των εργασιών. Ο αντικειμενοστραφής σχεδιασμός (**Object-Oriented design (OOD)**) καθορίζει όλους τους τύπους των αντικειμένων που είναι απαραίτητοι, δείχνει πώς τα αντικείμενα αλληλεπιδρούν για να ολοκληρωθούν οι εργασίες και τελειοποιεί τον ορισμό του κάθε τύπου αντικειμένου

έτσι ώστε να μπορεί να υλοποιηθεί σε μια συγκεκριμένη γλώσσα. Ο Αντικειμενοστραφής Προγραμματισμός (**Object-Oriented Programming (OOP)**) αποτελείται από γραπτές δηλώσεις σε μια γλώσσα προγραμματισμού για να καθορίσει τι κάνει κάθε τύπος αντικειμένου.

Ένα αντικείμενο είναι ένας τύπος πράγματος — ένας πελάτης ή ένας υπάλληλος ή ένας μαθητής, καθώς επίσης και ένα κουμπί ή ένα μενού. Κάποια πράγματα, όπως οι πελάτες, υπάρχουν τόσο έξω από το σύστημα (αληθινός πελάτης) όσο και χωριστά στο εσωτερικό του συστήματος (μια αναπαράσταση στον υπολογιστή του πελάτη). Η αντικειμενοστραφής ανάπτυξη πληροφοριακών συστημάτων χρησιμοποιεί ένα διάγραμμα κλάσεων για την εμφάνιση όλων των κλάσεων των αντικειμένων στο σύστημα.

Η αντικειμενοστραφής προσέγγιση έχει διάφορα βασικά πλεονεκτήματα, μεταξύ των οποίων είναι η φυσικότητα και η επαναχρησιμοποίηση. Η προσέγγιση είναι φυσική — ή διαισθητική — για τους ανθρώπους, επειδή έχουν την τάση να σκέφτονται τον κόσμο ως αντικείμενα. Είναι λιγότερο φυσικό για έναν άνθρωπο να σκεφτεί πολύπλοκες διαδικασίες που αναπτύχθηκαν σε διαδικαστικές γλώσσες προγραμματισμού. Επίσης, επειδή η αντικειμενοστραφής προσέγγιση περιλαμβάνει κλάσεις αντικειμένων και πολλά συστήματα στην οργάνωση τους χρησιμοποιούν τα ίδια αντικείμενα, αυτές οι κλάσεις μπορούν να χρησιμοποιηθούν ξανά και ξανά κάθε φορά που χρειάζονται. Για παράδειγμα, σχεδόν όλα τα συστήματα χρησιμοποιούν μενού, πλαίσια διαλόγου, παράθυρα και κουμπιά, αλλά επίσης και πολλά συστήματα εντός της ίδιας εταιρείας χρησιμοποιούν κλάσεις όπως πελάτης, προϊόν και τιμολόγιο που μπορούν να επαναχρησιμοποιηθούν. Σαφώς, η αντικειμενοστραφής προσέγγιση είναι αρκετά διαφορετική από την δομημένη προσέγγιση.

Πολλά συστήματα που αναπτύσσονται σήμερα συνδυάζουν παραδοσιακές και αντικειμενοστραφείς προσεγγίσεις. Για το λόγο αυτό, είναι σημαντικό η κάλυψη τόσο των παραδοσιακών/δομημένων προσεγγίσεων όσο και των αντικειμενοστραφών προσεγγίσεων.

## 2.9. Σύγχρονες Τάσεις στην Ανάπτυξη Συστήματος

Ένα πράγμα που δεν αλλάζει ποτέ στο πεδίο των πληροφοριών συστημάτων είναι ότι τα πράγματα αλλάζουν πάντα. Πάντα εμφανίζονται νέα εργαλεία και τεχνικές - μερικές φορές με μεγάλη δημοσιότητα και αναμονή - και οι αναλυτές πάντα ψάχνουν για νέους και καλύτερους τρόπους εργασίας. Οι τεχνικές και οι κύκλοι ζωής που συζητήθηκαν προηγουμένως είναι παραδείγματα των εν εξελίξει αλλαγών στις μεθοδολογίες ανάπτυξης συστημάτων. Σε αυτήν την ενότητα θα παρουσιαστούν μερικές σημαντικές τρέχουσες τάσεις στην ανάπτυξη του συστήματος. Κάθε μία από αυτές τις τάσεις θα μπορούσε να κυριαρχήσει στην ανάπτυξη των συστημάτων στο μέλλον.

### 2.9.1 Ενοποιημένη διαδικασία (UP)

Η **ενιαία διαδικασία (ΕΔ)** είναι μια αντικειμενοστραφής μεθοδολογία ανάπτυξης συστήματος που προσφέρεται από την εταιρεία λογισμικού IBM και προέρχεται από τούς τρεις υποστηρικτές της **Unified Modeling Language(UML)**: Grady Booch, James Rumbaugh και τον Ivar Jacobson. Αποτελεί την προσπάθειά τους να καθορίσουν μια ολοκληρωμένη μεθοδολογία που εκτός από την παροχή πολλών μοναδικών χαρακτηριστικών, χρησιμοποιεί την UML για τα μοντέλα του συστήματος. Αν και θα μάθετε πολλά για την UML επειδή πρόκειται για ένα πρότυπο μοντελοποίησης για την **αντικειμενοστραφή προσέγγιση (ΟΟ)**, η UP δεν είναι μία κλασσική ΟΟ μεθοδολογία ανάπτυξης.

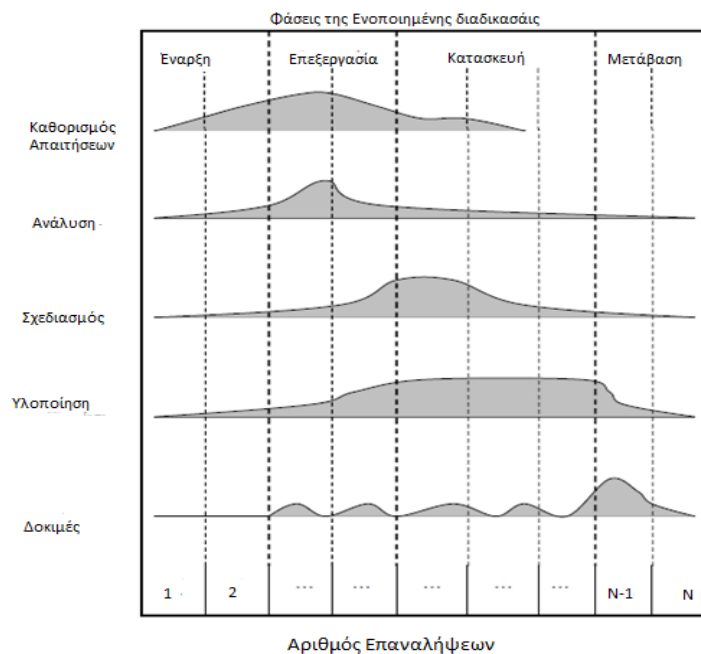
Η UP στοχεύει στην ενίσχυση των έξι "βέλτιστων πρακτικών" ανάπτυξης συστήματος που είναι κοινές σε πολλές μεθοδολογίες ανάπτυξης:

- Επαναληπτική ανάπτυξη
- Ορισμός και διαχείριση απαιτήσεων
- Χρήση συστατικών αρχιτεκτονικής
- Δημιουργία εικονικών μοντέλων
- Έλεγχος ποιότητας
- Έλεγχος αλλαγών

Η UP ορίζει τέσσερις φάσεις του κύκλου ζωής:

- έναρξη
- επεξεργασία
- κατασκευή
- μετάβαση

Ο κύκλος ζωής UP παρουσιάζεται στο σχήμα 2.9.



**Σχήμα 2.9:** Φάσεις της ενοποιημένης διαδικασίας

Η εναρκτήρια φάση ορίζει το πεδίο εφαρμογής του έργου καθορίζοντας περιπτώσεις χρήσης. Στην επόμενη φάση της επεξεργασίας, καθορίζονται οι περισσότερες από τις περιπτώσεις χρήσης με λεπτομέρειες και σχεδιάζεται η αρχιτεκτονική του συστήματος. Κατά την φάση της κατασκευής το προϊόν κατασκευάζεται, ενώ στη φάση της μετάβασης το προϊόν βρίσκεται σε κατάσταση δοκιμής.

### 2.9.2 Ακραίος Προγραμματισμός eXtreme Programming (XP)

Ο **ακραίος προγραμματισμός (XP)** είναι μία προσέγγιση ανάπτυξης συστήματος που πρόσφατα διαδόθηκε από τον Kent Beck. Στον XP προσαρμόζονται τεχνικές από πολλές πηγές και προσθέτονται μερικές νέες ιδέες. Μερικές φορές αναφέρεται ως μία "ελαφριά" μεθοδολογία ανάπτυξης συστήματος, δηλαδή διατηρείται απλή και επικεντρώνεται περισσότερο στην αποτελεσματικότητα της διαδικασίας ανάπτυξης. Είναι ένα παράδειγμα μιας σε μεγάλο βαθμό προσαρμοζόμενης προσέγγισης για τον SDLC.

Οι αναλυτές ξεκινούν τον προγραμματισμό του έργου αφήνοντας τους χρήστες να περιγράφουν ιστορίες χρηστών, που είναι παρόμοιες με περιπτώσεις χρήσης. Οι ιστορίες χρηστών είναι περιγραφές για το πώς θέλουν την υποστήριξη από το σύστημα — με άλλα λόγια περιγράφουν την απαιτούμενη λειτουργικότητα του συστήματος. Οι αναλυτές ετοιμάζουν ένα έγγραφο με αυτές τις ιστορίες με άτυπα περιγραφικά μοντέλα. Εκτός από τις ιστορίες του χρήστη, οι χρήστες περιγράφουν ένα σύνολο δοκιμών αποδοχής που θα αποδεικνύουν ότι το σύστημα παρέχει την απαιτούμενη λειτουργικότητα μετά την ολοκλήρωσή του.

Στη συνέχεια οι αναλυτές προγραμματίζουν μια σειρά εκδόσεων του έργου, όπου σε κάθε έκδοση συμπεριλαμβάνεται ένα μέρος του τελικού συστήματος, όπως ισχύει και με την προσέγγιση της σταδιακής ανάπτυξης. Το έργο προχωρά με την πρώτη έκδοση, που συνήθως παίρνει αρκετές επαναλήψεις για να ολοκληρωθεί. Όταν ολοκληρωθεί η πρώτη έκδοση, η δεύτερη έκδοση ξεκινά και το ίδιο συμβαίνει και με τις επόμενες εκδόσεις.

Ο XP είναι λίγο πολύ όπως και οι άλλες επαναληπτικές και επαυξητικές προσεγγίσεις. Όμως, ο XP περιέχει μερικά πρόσθετα χαρακτηριστικά γνωρίσματα που τον καθιστούν δημοφιλή. Για παράδειγμα απαιτεί συνεχή δοκιμή, συνεχή ολοκλήρωση και μεγάλη συμμετοχή των χρηστών. Απαιτεί επίσης, ο προγραμματισμός να γίνεται από ομάδες των δύο προγραμματιστών που εργάζονται μαζί σε ένα σταθμό εργασίας όταν γράφουν και δοκιμάζουν τον κώδικα. Αυτό και άλλα χαρακτηριστικά γνωρίσματα δίνουν έμφαση στην ανοιχτή και αποτελεσματική επικοινωνία μεταξύ των μελών της ομάδας. Ένα τελικό χαρακτηριστικό γνώρισμα είναι ότι οι προγραμματιστές δεν θα πρέπει να εργάζονται περισσότερο από 40 ώρες την εβδομάδα, για να αποδειχθεί ότι το σύστημα μπορεί να ολοκληρωθεί σύμφωνα με το χρονοδιάγραμμα χωρίς να απαιτούνται υπερβολικές ώρες εργασίας.

### 2.9.3 Μεθοδολογία SCRUM

Η **Scrum** είναι μία νέα μεθοδολογία ευέλικτης/προσαρμοστικής ανάπτυξης. Ο όρος Scrum αναφέρεται στο παιχνίδι του ράγκμπι. Το όνομα δόθηκε καθώς υπάρχουν πολλές ομοιότητες μεταξύ του αθλήματος και της προσέγγισης ανάπτυξης συστήματος. Και οι δύο είναι γρήγορες, προσαρμοστικές και αυτό-ρυθμιζόμενες. Η βασική ιδέα πίσω από την Scrum είναι ότι η μεθοδολογία πρέπει να ανταποκριθεί σε μια τρέχουσα κατάσταση γρήγορα και όσο το δυνατόν θετικότερα.

Η φιλοσοφία του scrum εστιάζεται στην γρήγορη ανταπόκριση σε αλλαγές ενός δυναμικού περιβάλλοντος στο οποίο οι χρήστες δεν γνωρίζουν ακριβώς τι χρειάζεται και μπορεί να αλλάζουν συχνά προτεραιότητες. Σε αυτό το είδος του περιβάλλοντος, οι αλλαγές είναι τόσες πολλές που τελικά το έργο τελματώνει και δεν φτάνει ποτέ σε ολοκλήρωση. Η scrum υπερέχει σε αυτές τις καταστάσεις έναντι των άλλων προσεγγίσεων επειδή επικεντρώνεται κατά κύριο λόγο στην ομάδα ανάπτυξης και στο έργο τους. Δίνει έμφαση στα άτομα περισσότερο από όσο στις διεργασίες και περιγράφει πώς οι ομάδες των προγραμματιστών μπορούν να εργαστούν μαζί για να οικοδομήσουν το λογισμικό σε μια σειρά από σύντομα υποέργα. Κλειδί σε αυτή η φιλοσοφία είναι ότι ασκεί τον απόλυτο έλεγχο μια ομάδα ειδικών.

#### *Ερωτήσεις - Δραστηριότητες - Θέματα προς συζήτηση*

1. Δημιουργήστε ένα σχεδιάγραμμα του δωματίου σας και στη συνέχεια προβείτε στην περιγραφή του. Αποτελούν και τα δύο μοντέλα του δωματίου σας; Που συναντάται περισσότερο η ακρίβεια και η λεπτομέρεια; Ποιο από τα δύο θα ακολουθούσε με ευκολία κάποιος που δεν σας έχει ξαναεπισκεφτεί;

2. Περιγράψτε μία «τεχνική» που θα σας βοηθήσει να ολοκληρώσετε την δραστηριότητα «Είμαι στην ώρα μου στο μάθημα». Ποια εργαλεία θα μπορούσατε να χρησιμοποιήσετε;
3. Περιγράψτε μία «τεχνική» που θα σας βοηθήσει να ολοκληρώσετε την δραστηριότητα «Παραδίδω τις εργασίες έγκαιρα». Ποια εργαλεία θα μπορούσατε να χρησιμοποιήσετε;
4. Ποιες άλλες τεχνικές χρησιμοποιείτε για να ολοκληρώσετε τις καθημερινές σας δραστηριότητες;
5. Υπάρχουν τουλάχιστον δύο προσεγγίσεις για την ανάπτυξη συστήματος, μια ποικιλία των κύκλων ζωής ανάπτυξης συστήματος και ένας μακρύς κατάλογος τεχνικών και μοντέλων που χρησιμοποιούνται σε ορισμένες προσεγγίσεις και σε άλλες όχι. Σκεφτείτε γιατί συμβαίνει αυτό. Συζητήστε τους παρακάτω πιθανούς λόγους και προσδιορίστε ποιός είναι πιο σημαντικός τεκμηριώνοντας την άποψή σας: το πεδίο εφαρμογής είναι αρκετά νέο; η τεχνολογία αλλάζει πολύ γρήγορα; διαφορετικοί οργανισμοί έχουν διαφορετικές ανάγκες; υπάρχουν πολλά διαφορετικά είδη συστημάτων; στην ανάπτυξη συστημάτων εμπλέκονται άτομα με διαφορετικά υπόβαθρα;
6. Περιηγηθείτε στο διαδίκτυο για να βρείτε εταιρείες ανάπτυξης λογισμικού. Αναζητήστε την προσέγγιση που ακολουθούν στην ανάπτυξη συστήματος καθώς και τα χρησιμοποιούμενα εργαλεία. Περιγράφουν τον Κύκλο Ζωής Ανάπτυξης Συστήματος; Κάνουν αναφορά σε IDE ή σε εργαλεία οπτικής μοντελοποίησης;
7. Θεωρείστε ότι ξεκινάτε τις σπουδές σας στην Α' ΕΠΑΛ. Στόχος σας είναι να ολοκληρώσετε τις σπουδές σας στο προβλεπόμενο χρονικό διάστημα αποκτώντας εκείνα τα εφόδια που θα σας βοηθήσουν στην μετέπειτα εξέλιξή σας. Αποφασίζετε να αναπτύξετε μία μεθοδολογία «ολοκλήρωσης της εκπαίδευσης». Στο πλαίσιο αυτό προσπαθείτε να απαντήσετε στα παρακάτω ερωτήματα: Ποια είναι τα στάδια του προσωπικού σας κύκλου ζωής ολοκλήρωσης της εκπαίδευσης; Ποιες δραστηριότητες περιέχονται σε κάθε στάδιο; Ποιες τεχνικές, μοντέλα και εργαλεία θα μπορούσατε να χρησιμοποιήσετε για να υλοποιήσετε τις παραπάνω δραστηριότητες;
8. Σας έχει ανατεθεί η ανάπτυξη του συστήματος αυτοματοποίησης παρακολούθησης των εργασιών που βρίσκονται σε εξέλιξη καθώς και του τελικού αποθέματος εμπορευμάτων. Έρχεστε αντιμέτωποι με μία νέα πρόκληση. Πώς θα αντιμετωπίσουν οι εργαζόμενοι τις νέες αλλαγές; Από την άλλη μεριά χρειάζεστε την συνεργασία των εργαζομένων όσον αφορά το κομμάτι της τεχνολογίας. Για το δεύτερο κομμάτι αυτό της λογιστικής γνωρίζετε αρκετά πράγματα. Μπορείτε να διακρίνετε ποια είναι τα δύο συστήματα που θα κληθείτε να αναπτύξετε; Ποιες παραλλαγές του Κύκλου Ζωής Ανάπτυξης Συστήματος είναι πιο κατάλληλες για το υπάρχον πλαίσιο; Θα επιλέγατε τις προσεγγίσεις UP, XP ή Scrum; Σε ποιες



δραστηριότητες της ανάλυσης και του σχεδιασμού θα έπρεπε να συμμετέχουν οι εργαζόμενοι και τα στελέχη;

9. Επιλέξτε ένα επιχειρησιακό έργο της προτίμησής σας για να το μηχανογραφήσετε. Πώς θα μπορούσατε να χρησιμοποιούσατε τα στάδια του Κύκλου Ζωής Ανάπτυξης Συστήματος για να το υλοποιήσετε. Παραθέστε ενδεικτικά παραδείγματα.
10. Ποια από τις δύο προσεγγίσεις για την ανάπτυξη του συστήματος είναι η πιο πρόσφατη;
11. Δώστε μερικά χαρακτηριστικά της ενοποιημένης διαδικασίας (UM).
12. Συζητήστε τα πλεονεκτήματα του ακραίου προγραμματισμού (XP).
13. Δώστε μερικά χαρακτηριστικά της προσέγγισης Scrum

### Μελέτη Περίπτωσης

#### Σχολική Βιβλιοθήκη

Η Βιβλιοθήκη του σχολείου σας είναι αρκετά μεγάλη, έχει πολλά βιβλία σχολικά, λογοτεχνικά, ιστορικά, ποιήματα κ.α. Παρόλα αυτά δεν σας έχει απασχολήσει ποια βιβλία περιλαμβάνονται καθώς δεν είναι εύκολη η πρόσβαση σε αυτά. Σκέφτεστε ότι μια σχολική βιβλιοθήκη στην σύγχρονη εποχή θα έπρεπε να περιλαμβάνει και e-books, e-magazines και γενικότερα να υπάρχει πρόσβαση μέσω εφαρμογών στα περιεχόμενα βιβλίων, στους τίτλους των βιβλίων καθώς και δυνατότητα δανεισμού. Θα ήταν ωραίο όταν χρησιμοποιείτε τον υπολογιστή ή το κινητό σας να μπορείτε να βρείτε το βιβλίο που χρειάζεστε, να μάθετε περισσότερες πληροφορίες για αυτό και να έχετε τη δυνατότητα να το δανειστείτε. Επίσης, να μπορείτε να κατεβάσετε τμήματα ή και ολόκληρα ψηφιακά βιβλία που χρειάζεστε στα μαθήματά σας.

Ωρα να ξεκινήσετε την ανάλυση του έργου. Είστε οι **αναλυτές του συστήματος** που θα υλοποιηθεί. Σκεφτείτε αυτά που μάθατε σε αυτό το κεφάλαιο (τον κύκλο ζωής λογισμικού) και αναλογιστείτε τις ενέργειες που πρέπει να γίνουν. Χωριστείτε σε ομάδες. Κάθε ομάδα θα αναλάβει να επεξεργαστεί μια προσέγγιση ανάπτυξης συστήματος από αυτές που μάθατε. Παρουσιάστε τις προτάσεις σας στις υπόλοιπες ομάδες. Στη συνέχεια η ολομέλεια της τάξης θα αποφασίσει ποια προσέγγιση θα ακολουθηθεί στην ανάπτυξη του συστήματος σχολικής βιβλιοθήκης.

#### *Βιβλιογραφία*

Στα Ελληνικά

Βεσκούκης, Β.(2000) *Τεχνολογία Λογισμικού Ι*. Πάτρα: Ε.Α.Π.

Στα Αγγλικά

O'Brien A. J. & Marakas. G. (2008). *Introduction to Information Systems*, 14th/edition, New York: McGraw-Hill.

Satzinger, J. W. & Jackson B. R., Burd D. S. (2009). *Systems Analysis and Design in a Changing World*, Boston: MA: Thomson Course Technology.